AMSAA

TECHNICAL REPORT NO. 279

A COMPUTER PROGRAM FOR ESTIMATION OF PARAMETERS OF THE WEIBULL INTENSITY FUNCTION AND FOR THE CRAMER-VON MISES GOODNESS OF FIT TEST

EDWARD F. BELBOT

JULY 1979

U. S. ARMY MATERIEL SYSTEMS ANALYSIS ACTIVITY
ABERDEEN PROVING GROUND, MARYLAND

DISPOSITION

Destroy this report when no longer needed. Do not return it to the originator.

DISCLAIMER

The findings in this report are not to be construed as an official Department of the Army position unless so specified by other official documentation.

WARNING

Information and data contained in this document are based on the input available at the time of preparation. The results may be subject to change and should not be construed as representing the DARCOM position unless so specified.

TRADE NAMES

The use of trade names in this report does not constitute an official endorsement or approval of the use of such commercial hardware or software. The report may not be cited for purposes of advertisement.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER Technical Report No. 279	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE *(and Subtitle)* A Computer Program for Estimation of Parameters of the Weibull Intensity Function and for the Cramer-Von Mises Goodness of Fit Test		5. TYPE OF REPORT & PERIOD COVERED 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Edward F. Belbot		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS U S Army Materiel Systems Analysis Activity Aberdeen Proving Ground, MD 21005		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS DA Project No. 1R765706M541
11. CONTROLLING OFFICE NAME AND ADDRESS Commander, U S Army Materiel Development and Readiness Command, Alexandria, VA 22333		12. REPORT DATE July 1979 13. NUMBER OF PAGES 70
14. MONITORING AGENCY NAME & ADDRESS*(if different from Controlling Office)*		15. SECURITY CLASS. *(of this report)* Unclassified 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT *(of this Report)*

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT *(of the abstract entered in Block 20, if different from Report)*

18. SUPPLEMENTARY NOTES

19. KEY WORDS *(Continue on reverse side if necessary and identify by block number)*

Weibull Intensity Function
Goodness of Fit Test
FORTRAN
Iterative Procedure

20. ABSTRACT *(Continue on reverse side if necessary and identify by block number)*

This report describes the structure and use of a digital computer program written in Standard FORTRAN, which can be used to obtain the maximum likelihood estimates of the parameters of the Weibull intensity function. The program also performs the Cramer-Von Mises goodness of fit test.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

CONTENTS

LIST OF FIGURES

A COMPUTER PROGRAM FOR ESTIMATION OF PARAMETERS OF THE WEIBULL INTENSITY FUNCTION AND FOR THE CRAMER-VON MISES GOODNESS OF FIT TEST

1. INTRODUCTION

The Weibull intensity function

$$U(X) = \lambda\beta X^{\beta-1} \tag{1.1}$$

$\lambda>0$, $\beta>0$, $X>0$, is frequently used as a model for the determination of reliability growth and wear-out characteristics for a wide variety of complex, repairable systems. The failure rates of military equipment such as vehicles, aircraft, guided missiles, electronic computer systems, and ammunition are being evaluated using this model.

Formulas have been developed by the Army Materiel Systems Analysis Activity (AMSAA) for the maximum likelihood estimation of the unknown parameters λ and β, based upon sample data. These estimation formulas for the Weibull process, found in Crow (1975), can be stated as

$$\hat{\beta} = \frac{\sum_{i=1}^{K} N_i}{\hat{\lambda} \sum_{i=1}^{K} (T_{i2}^{\hat{\beta}} \ln T_{i2} - T_{i1}^{\hat{\beta}} \ln T_{i1}) - \sum_{i=1}^{K} \sum_{j=1}^{N_i} \ln X_{ij}} \tag{1.2}$$

$$\hat{\lambda} = \frac{\sum_{i=1}^{K} N_i}{\sum_{i=1}^{K} (T_{i2}^{\hat{\beta}} - T_{i1}^{\hat{\beta}})} \tag{1.3}$$

where:

K is the number of systems under study;
N_i is the total number of failures (or occurrences of an event under study, such as unscheduled maintenance actions; etc.) for the i^{th} system;
T_{i1} is the starting time of the period of continuous observation of the i^{th} system;
T_{i2} is the ending time of the period of continuous observation of the i^{th} system;
X_{ij} is the j^{th} time of occurrence of the failure (or event), for the i^{th} system;

ln is the natural logarithm, and $0 \cdot \ln(0)$ is defined to be 0.

To expedite the computation of these estimates, AMSAA developed a FORTRAN computer program to calculate $\hat{\beta}$ and $\hat{\lambda}$. That program, which was documented in Belbot (1974), was successfully employed by the U S Army Materiel Development and Readiness Command (DARCOM), various subordinate commands and several project managers' offices, as well as by AMSAA. After the parameters were determined by that program, a goodness of fit test was frequently used to test statistically the hypothesis that the failure times of the systems being analyzed followed a nonhomogeneous Poisson process with Weibull intensity function (see Crow [1975]). The modified Cramér-Von Mises goodness of fit statistic was computed, either by hand or by a separate computer program.

Obviously, the consolidation of an automated goodness of fit test with the computer routine which estimates the parameters β and λ, would increase efficiency and accuracy. Either the manual calculations or the use of a separate program would be eliminated. To accomplish properly this consolidation, the estimation procedure was subordinated to a new main program which also controls the input of data and the execution of the goodness of fit test. Because of the radical nature of this redesign, it was appropriate to incorporate other new features at the same time. Principal among these new features are a simplified input procedure and dynamic data storage allocation. The resulting computer program is easier to use and provides more information that its antecedent program. This note will explain the structure and the use of this new program.

2. COMPUTING PROCEDURE

2.1 Estimation of Parameters

Since the formulas (1.2) and (1.3) do not, in general, yield $\hat{\beta}$ and $\hat{\lambda}$ in closed form, an iterative technique is required. Formula (1.2) may be recast as

$$\frac{\sum_{i=1}^{K}\sum_{j=1}^{N_i} \ln X_{ij}}{\sum_{i=1}^{K} N_i} - \frac{\sum_{i=1}^{K} \left(T_{i2}^{\hat{\beta}} \ln T_{i2} - T_{i1}^{\hat{\beta}} \ln(T_{i1})\right)}{\sum_{i=1}^{K} \left(T_{i2}^{\hat{\beta}} - T_{i1}^{\hat{\beta}}\right)} - \frac{1}{\hat{\beta}} = 0 \qquad (2.1)$$

by replacing $\hat{\lambda}$ by its equivalent expression from equation (1.3), and by execution of a few simple algebraic operations. Equation (2.1) now consists of a constant with regard to $\hat{\beta}$, minus a function of $\hat{\beta}$, yielding 0, or simply

$$A - D(\hat{\beta}) = 0 \qquad (2.2)$$

The correct value of $\hat{\beta}$ will satisfy equation (2.2) and can be used to calculate the corresponding value of $\hat{\lambda}$.

The solution for $\hat{\beta}$ is iteratively determined in the following way. For an initial estimate $\hat{\beta}'$ which is assumed to be larger than the true $\hat{\beta}$, the expression A-D($\hat{\beta}'$) is evaluated. For all values of $\hat{\beta}'$ larger than the true $\hat{\beta}$, the subtraction yields a negative result. After each negative result, $\hat{\beta}'$ is reduced by the initial step size of 1, and A-D($\hat{\beta}'$) is again evaluated.

When a positive number results from the subtraction, indicating that $\hat{\beta}'$ is too small, the step size is decreased to 0.10 of the present step size, the previous value of $\hat{\beta}'$ which gave a negative result for A-D($\hat{\beta}'$), is adjusted by the new step size and the evaluation process begins again.

The iteration procedure continues, adjusting $\hat{\beta}'$ by the new step sizes, until the left side of equation (2.2) is within a specified tolerance ε of 0. $\hat{\lambda}$ is then calculated based on $\hat{\beta}$, using factors already computed in finding $\hat{\beta}$. This procedure is summarized by the state diagram (Figure 2.1).

2.2 Goodness of Fit Test

The Cramer-Von Mises Goodness of Fit Test is appropriate whenever the starting time of each system is equal to 0. To perform this test, the program first transforms the failure times. For time truncated testing, the failure times for each system are divided by the ending time of the test period for that system. In failure truncated testing, for every system, all the failure times except the last, are divided by the last failure time. The last failure time is thereafter excluded from the calculations and the number of transformed failures is one less than the original number of failures for each such system. All the transformed failure times are then sorted into increasing order.

Next, the unbiased estimate $\bar{\beta}$ of the estimated shape parameter $\hat{\beta}$, is calculated using the relation:

$$\bar{\beta} = \frac{M-1}{N} \hat{\beta} \qquad (2.3)$$

where:

M is the number of transformed failure times;
and N is the number of original failure times.

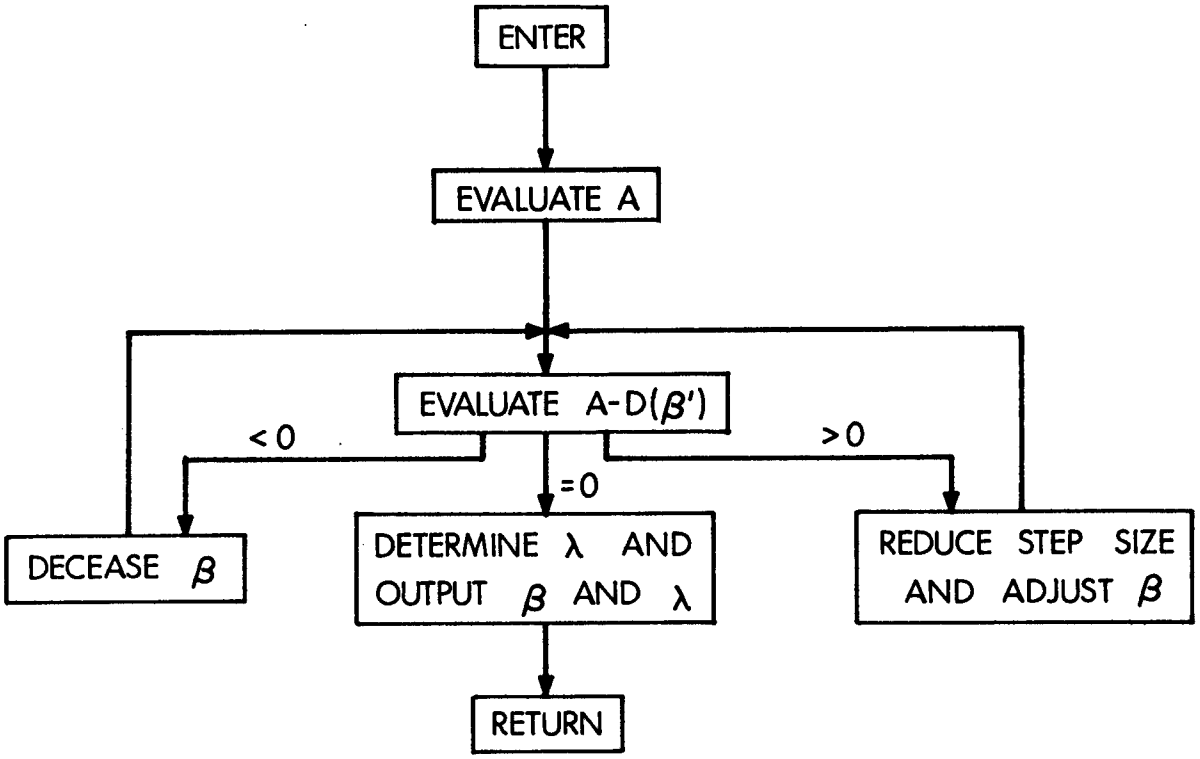

Figure 2.1 State Diagram

Finally, the Cramer-Von Mises statistic C_M^2 is computed by the formula:

$$C_M^2 = \frac{1}{12M} + \sum_{i=1}^{M} \left(Z_i^{\bar{\beta}} - \frac{2i-1}{2M}\right)^2 \qquad (2.4)$$

where the Z_i are the transformed failure times. An explanation of this test and a table of critical values of C_M^2 may be found in Crow (1975).

Next page is blank.

3. DESCRIPTION OF PROGRAM

3.1 Major Features

The program, which is listed in Appendix A, has some important features. First, the program is written in American National Standards Institute (ANSI) FORTRAN X3.9-1966, and should therefore execute on any computer having a compiler for this language. Secondly, the amount of storage required to use this program should not cause difficulties since all data arrays are dynamically allocated under control of the main program (see Chung-Phillips, et al., [1975]).

Finally, and perhaps most importantly for the user, this program uses free-field input, that is, no specific format is required for the input data. The time and the effort regularly expended in preparing data for input, are greatly reduced because of this feature. Moreover, the misalignment of data fields to formats, a frequent source of errors in using many computer programs, is eliminated entirely. While no input scheme can be regarded as foolproof, free-field input is much more flexible than fixed-field format specifications.

3.2 Overall Characteristics

All calculations in the program are made in double precision mode. Experience has shown that the use of single precision variables for these calculations often results in significant discrepancies in the estimates of the parameters due to errors accumulated during the iterative process.

The modular structure of the program (see Figure 3.1) reflects organization by functional purpose. The input of data, certain intermediate calculations, the estimation of parameters, and the goodness of fit test are each performed by an independent module. Major subprograms print their results as the values become available. Subroutines which detect errors, print diagnostic messages naming the detecting routine and briefly stating the difficulty, and then attempt to continue processing when possible. Independence of the subroutines is maintained by restricting communication between individual subprograms to the passage of formal parameters in argument lists. No COMMON statements are used.

3.3 Specific Details of Routines

In addition to controlling the major modules, the main program also allocates storage for data arrays, as stated in Section 3.1. The allocation is based upon the maximum number of failures, NFAIL, and the maximum number of systems, NSYS. The master data array, BLOCK, has a length of NTOTAL, equal to the value of NFAIL plus six times the value of NSYS. To redimension the entire program, one merely adjusts the values of NFAIL, NSYS and NTOTAL in the DATA statements at the beginning of the main routine and changes the size of the array BLOCK, also found at the beginning of the main program, to equal the new value of NTOTAL.

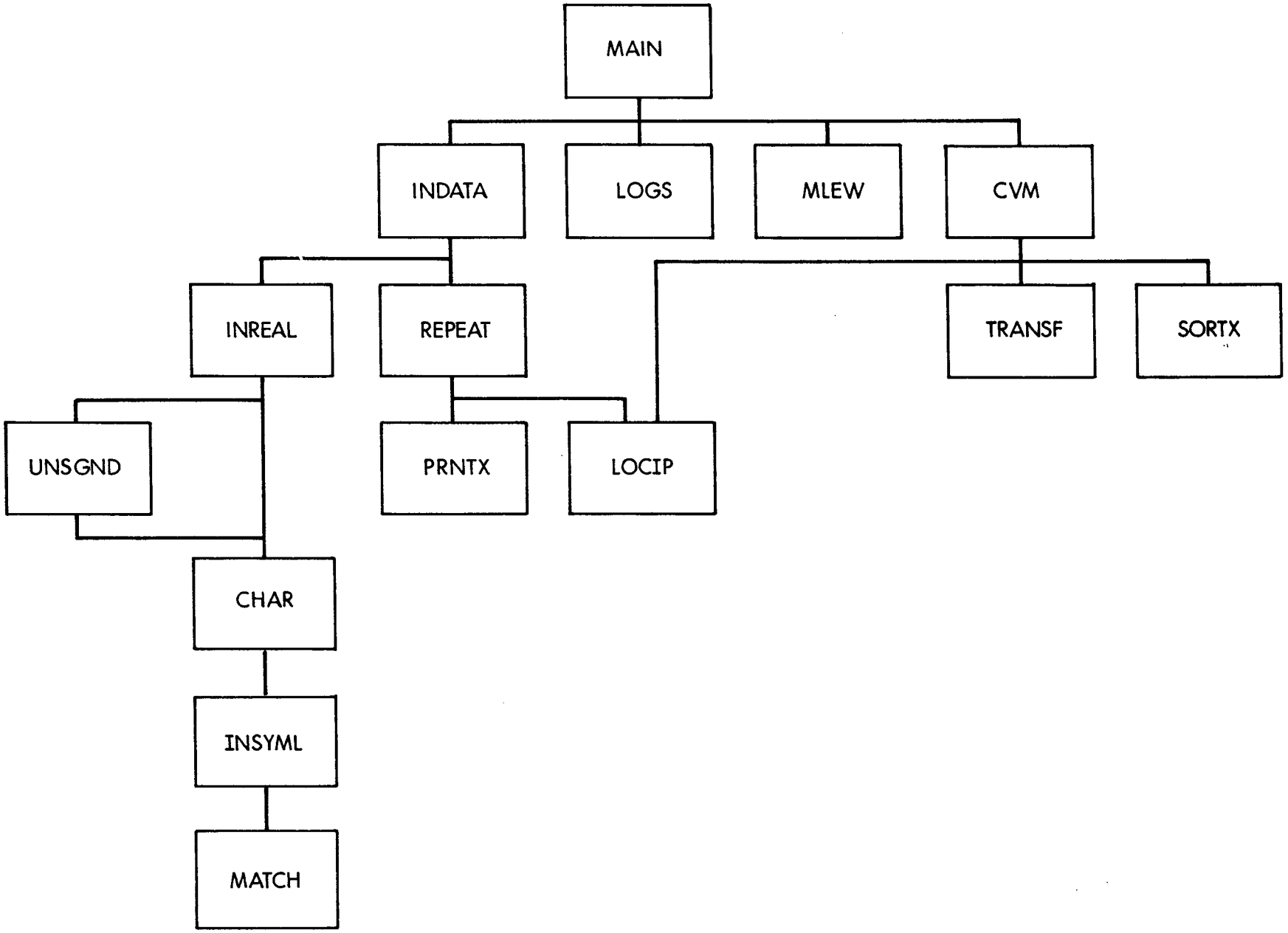

Figure 3.1 Module Organization Chart.

By these actions, all data arrays in all subroutines will be properly resized. Since the program size excluding data arrays, is less than 5,000 words, computer memory requirements can be scaled to problem size through use of this feature.

Also found in the DATA statements at the beginning of the main routine are the unit number for input, IUNIT, and the logical switch ECHO which controls the printing of the input data. Just as the storage allocation values, these values may be changed as needed.

The first major module is for the input of data. The INDATA module reads the beginning and the ending times and the failure times from the input unit, IUNIT. If the logical variable ECHO is true, the submodule REPEAT will print the input, using the LOCIP subroutine to isolate in storage the failures for each system and the PRNTX subroutine to print them. The failure times are stored in a linear array with negative signs appended to the failures associated with even numbered systems. This scheme preserves the identification of the failures with the respective systems without using additional storage.

The free-field reading of data is performed by the INREAL submodule. This submodule, consisting of the routines INREAL, CHAR, INSYML, MATCH and UNSGND, is a translation from ALGOL into FORTRAN of Algorithm 239 of the Association for Computing Machinery (see McKeeman [1964]). Specific details concerning the input arrangement are given in Section 4.

The INDATA subroutine also sets three logical variables depending on the input. If data errors are encountered, the variable NOGOOD is made true. If all systems start at 0.0, then the goodness of fit test will be appropriate and so the variable GOF is set to true. Lastly, when the end of input is reached, the logical variable HALT is returned as true.

The second principal module, LOGS, calculates the logarithms of the beginning and the ending times, and the sum of the logarithms of the failure times. For computational purposes, a beginning time of 0.0 is defined to have a logarithm of 0.0 instead of infinity. Note that failure times of 0.0 are not valid for this model and are flagged as errors by the preceding INDATA module.

The next major module, MLEW, computes the maximum likelihood estimates of the parameters of the Weibull intensity function for the given data, using the formulas discussed in Section 2.1. If unsuccessful, this module will report one of three possible error conditions. The first error message "BETA LESS THAN 0.00000001" indicates that the data should be rechecked for the reasonableness of a very small $\hat{\beta}$. The second error message is "INITIAL ESTIMATE OF BETA IS TOO SMALL." Since the initial estimate of BETA is set to 10 at the start of the MLEW subprogram, this message indicates some pecularity of the data. (In general, $0<\hat{\beta}<10$.) The third message, "STEP-SIZE HAS BECOME INSIGNIFICANT - BETA NOT RESOLVABLE," indicates that the module has gone as far as

possible trying to meet the tolerance set for the difference $A\text{-}D(\hat{\beta}')$. This tolerance, EPSILN, may be enlarged by changing the assignment statement also located at the beginning of the MLEW subprogram.

The last major module, CVM, performs the Cramér-Von Mises goodness of fit test, as described in Section 2.2. The failure times for each system are located in storage using the LOCIP subroutine, and examined to determine if the testing was time truncated or failure truncated. The failure times are then transformed by the TRANSF subroutine and sorted by the SORTX subroutine. (SORTX is a modification of an utility subprogram described in Campbell, et al., [1970].) The unbiased estimate of β, UNBETA, is calculated next, as explained earlier. The last phase depends on the system starting times. If any starting time is non-zero, the module terminates with a message stating that the Cramér-Von Mises goodness of fit test is not appropriate. Otherwise, the goodness of fit statistic, CM2, is computed and printed.

4. INPUT REQUIREMENTS

As stated previously, the input for this program is free-field. The only requirement regarding spacing is that at least one blank column separate adjacent values. The values must not run together. This means that the program generally takes the same view of the data that a person would, namely, that each cluster of numeric characters constitutes one data value. The only exception to this rule occurs at the boundaries of records. Since the input is treated as a continuous stream, a string of characters beginning in the first column of a record, is considered a continuation of the string of characters ending in the last column of the previous record, if any. Record boundaries are not delimiters; blanks are the only delimiters.

The data required for this program consist of the beginning and ending times of the test period for each system, and the failure times for each system. The arrangement of the input, (which is also stated in the comments of the INDATA subroutine), is by system. The first data value is the beginning time of the first system. The second data value is the ending time of the first system. Next is the failure times for the first system, followed by a negative value to mark the end of the first system. The same pattern, beginning time, ending time, failure times, and negative trailer, is repeated for each subsequent system in the first data case.

Another negative value (making two in a row), signals the end of input of the current data case, and the beginning of the computational procedures. The same arrangement may be repeated for as many cases as desired per program run. When the input routine encounters a negative value after completing a case, (that is, the third negative value in a row), the end of the program run is indicated.

Thus, as a simple example, if one desired to use this program for one run consisting of one case wherein one system experienced seven failures, the input data would be: the beginning time, the ending time, the seven failure times, and three negative values. To demonstrate the latitude of the input requirements, the data for a number of test cases are shown in Appendix B. Notice that any negative value is acceptable as a trailer and that data values may be entered with or without decimal points. Although not shown in the examples, data values may also be in exponential form, that is, containing 'E,' '+' or '-.'

Next page is blank.

5. TRANSFERABILITY AND MODIFICATION

Since this program is written in standard FORTRAN, transfer to other computer systems should be straightforward. To assist in the transfer process, Appendix C contains the output produced by the program for the input shown in Appendix B. This output was generated on a Control Data Corporation (CDC) Cyber 76 Computer, using the program exactly as listed in Appendix A. (Note that non-standard PROGRAM statement required by the CDC Cyber.)

The input for these test cases came from records of eighty characters each. If the input record length is other than eighty, two changes may be required. The value of the variable LENGTH and, if necessary, the dimensioned size of the array BUFFER, should be adjusted in the INSYML subroutine.

Alternatively, one could replace the entire INREAL submodule. Although these routines were written to be fully transportable, running time might be saved by using the system defined free-field reading capability of any computer having such a feature. As an example, a substitute for the INREAL submodule, suitable for the CDC Cyber 76, is shown in Figure 5.1. Such substitutes, however, are system dependent and not readily transferable.

Next page is blank.

FIGURE 5.1 SUBSTITUTE FOR INREAL SUBMODULE

```
SUBROUTINE INREAL (IU, X)
DOUBLE PRECISION X
READ (IU, *) X
RETURN
END
```

Next page is blank.

REFERENCES

1. Belbot, E. F., A Computer Program for Estimation of Parameters of the Weibull System Failure Rate Function, U S Army Materiel Systems Analysis Activity Technical Report No. 99, 1974.

2. Campbell, L. W. and Beck, G. A., BRLESC I/II FORTRAN, U S Army Aberdeen Research and Development Center Technical Report No. 5, 1970.

3. Chung-Phillips, A. and Rosen, R. W., "A Note on Dynamic Data Storage in FORTRAN IV," Computer Journal, Volume 18, No. 4, 1975.

4. Crow, L. H., Reliability Analysis for Complex, Repairable Systems, U S Army Materiel Systems Analysis Activity Technical Report No. 138, 1975.

5. McKeeman, W. M., "Algorithm 239 - Free Field Read," Communications of the Association for Computing Machinery, Volume 8, 1964.

6. USA Standard FORTRAN (USAS X3.9 - 1966). USA Standards Institute (now designated the American National Standards Institute, Inc.), 1966.

Next page is blank.

APPENDIX A

LISTING OF PROGRAM

Next page is blank.

```
      PROGRAM MAIN (INPUT=/80,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT)          MAIN  10
C        THE ABOVE STATEMENT IS NON-STANDARD, BUT REQUIRED FOR          MAIN  20
C        CDC FORTRAN.                                                   MAIN  30
C-                                                                      MAIN  40
C-                                                                      MAIN  50
C-              MAIN DRIVER FOR WEIBULL INTENSITY MODEL PARAMETER       MAIN  60
C-           ESTIMATION AND GOODNESS OF FIT TEST.                       MAIN  70
C-                                                                      MAIN  80
C-              (VERSION OF 27 JULY 1979)                               MAIN  90
C-                                                                      MAIN 100
C-              INPUT REQUIREMENTS ARE DESCRIBED IN THE 'INDATA'        MAIN 110
C-           SUBROUTINE.                                                MAIN 120
C-                                                                      MAIN 130
C                                                                       MAIN 140
      DOUBLE PRECISION BETA, SUMLNX                                     MAIN 150
      DOUBLE PRECISION BLOCK(11000)                                     MAIN 160
      LOGICAL ECHO, FAULT, GOF, HALT, NOGOOD                            MAIN 170
C                                                                       MAIN 180
      DATA IUNIT /5/                                                    MAIN 190
      DATA ECHO /.TRUE./                                                MAIN 200
      DATA NFAIL, NSYS /5000, 1000/                                     MAIN 210
      DATA NTOTAL /11000/                                               MAIN 220
C                                                                       MAIN 230
      WRITE (6,20)                                                      MAIN 240
C                                                                       MAIN 250
C        ALLOCATE STORAGE BASED ON MAXIMUM NUMBERS OF FAILURES          MAIN 260
C             AND SYSTEMS.                                              MAIN 270
      NS2=NSYS*2                                                        MAIN 280
      I1=1                                                              MAIN 290
      I2=I1+NFAIL                                                       MAIN 300
      I3=I2+NS2                                                         MAIN 310
      I4=I3+NS2                                                         MAIN 320
      ITOTAL=I4+NS2-1                                                   MAIN 330
      IF (ITOTAL.LE.NTOTAL) GO TO 10                                    MAIN 340
      WRITE (6,30) ITOTAL,NTOTAL                                        MAIN 350
      STOP                                                              MAIN 360
C                                                                       MAIN 370
C        BEGIN PROCESSING.                                              MAIN 380
   10 CALL INDATA (BLOCK(I1),BLOCK(I2),NOGOOD,GOF,HALT,NFAIL,NSYS,M,K,IUMAIN 390
     1NIT,ECHO)                                                         MAIN 400
      IF (HALT) STOP                                                    MAIN 410
      IF (NOGOOD) GO TO 10                                              MAIN 420
      CALL LOGS (BLOCK(I1),BLOCK(I2),BLOCK(I4),NFAIL,NSYS,M,K,SUMLNX)   MAIN 430
      CALL MLEW (SUMLNX,BLOCK(I2),BLOCK(I3),BLOCK(I4),FAULT,NSYS,M,K,BETMAIN 440
     1A)                                                                MAIN 450
      IF (.NOT.FAULT) CALL CVM (BLOCK(I1),BLOCK(I2),NFAIL,NSYS,M,K,BETA,MAIN 460
     1GOF)                                                              MAIN 470
      GO TO 10                                                          MAIN 480
C                                                                       MAIN 490
   20 FORMAT (71H1 WEIBULL INTENSITY MODEL PARAMETER ESTIMATION AND GOODMAIN 500
     1NESS OF FIT TEST/25HO VERSION OF 27 JULY 1979////)                MAIN 510
   30 FORMAT (33H1 AMOUNT OF STORAGE REQUESTED IS ,I6,40H WORDS.  AMOUNTMAIN 520
     1 OF STORAGE AVAILABLE IS ,I5,25H WORDS.  PROGRAM ABORTED./)       MAIN 530
      END                                                               MAIN 540
```

```
      SUBROUTINE CVM (X,C,NFAIL,NSYS,M,K,BETA,GOF)                      CVM   10
COMMENT       THIS SUBROUTINE PERFORMS THE CRAMER-VON MISES GOODNESS     CVM   20
C        OF FIT TEST.                                                   CVM   30
C                                                                       CVM   40
      DOUBLE PRECISION X(NFAIL), C(NSYS,2)                              CVM   50
      DOUBLE PRECISION BETA, CM2, DM, SUMSQS, TQ, TWOM, UNBETA          CVM   60
      DOUBLE PRECISION TERM, TERM1, TERM2                               CVM   70
      LOGICAL GOF                                                       CVM   80
      LOGICAL TIMETR                                                    CVM   90
C                                                                       CVM  100
      WRITE (6,60) BETA,M                                               CVM  110
      IP=0                                                              CVM  120
      IBT=1                                                             CVM  130
      TIMETR=.TRUE.                                                     CVM  140
      N=M                                                               CVM  150
C                                                                       CVM  160
      DO 30 J=1,K                                                       CVM  170
      IB=IP+1                                                           CVM  180
      CALL LOCIP (X,NFAIL,N,IP)                                         CVM  190
      IE=IP                                                             CVM  200
      DO 10 I=IB,IE                                                     CVM  210
C        CHECK FOR FAILURE TRUNCATED TESTING.                           CVM  220
      IF (DABS(DABS(X(I))-C(J,2)).LE.1.0D-08) TIMETR=.FALSE.            CVM  230
   10 CONTINUE                                                          CVM  240
      IF (TIMETR) GO TO 20                                              CVM  250
      IE=IE-1                                                           CVM  260
      M=M-1                                                             CVM  270
      TIMETR=.TRUE.                                                     CVM  280
   20 CONTINUE                                                          CVM  290
      TQ=C(J,2)                                                         CVM  300
      IET=IE                                                            CVM  310
C        TRANSFORM THE FAILURES.                                        CVM  320
      CALL TRANSF (X,TQ,IB,IBT,IET,NFAIL)                               CVM  330
      IBT=IET+1                                                         CVM  340
   30 CONTINUE                                                          CVM  350
C                                                                       CVM  360
      DO 40 I=1,M                                                       CVM  370
      X(I)=DABS(X(I))                                                   CVM  380
   40 CONTINUE                                                          CVM  390
C        SORT THE TRANSFORMED FAILURES INTO INCREASING ORDER.           CVM  400
      CALL SORTX (X,M)                                                  CVM  410
C                                                                       CVM  420
      DM=DBLE(FLOAT(M))                                                 CVM  430
      TWOM=2.0D0*DM                                                     CVM  440
C        UNBIASED ESTIMATE OF BETA.                                     CVM  450
      UNBETA=BETA*(DM-1.0D0)/DBLE(FLOAT(N))                             CVM  460
      WRITE (6,110) UNBETA                                              CVM  470
      IF (GOF) GO TO 50                                                 CVM  480
      WRITE (6,100)                                                     CVM  490
      GO TO 70                                                          CVM  500
C                                                                       CVM  510
   50 SUMSQS=0.0D0                                                      CVM  520
      DO 60 I=1,M                                                       CVM  530
      TERM1=X(I)**UNBETA                                                CVM  540
      TERM2=DBLE(FLOAT(2*I-1))/TWOM                                     CVM  550
      TERM=TERM1-TERM2                                                  CVM  560
      SUMSQS=SUMSQS+TERM**2                                             CVM  570
```

```
   60 CONTINUE                                                          CVM  580
C        CRAMER-VON MISES STATISTIC.                                    CVM  590
      CM2=SUMSQS+(1.0D0/(12.0D0*DM))                                    CVM  600
      WRITE (6,90) CM2,M                                                CVM  610
C                                                                       CVM  620
   70 RETURN                                                            CVM  630
C                                                                       CVM  640
   80 FORMAT (42H1 CRAMER - VON MISES GOODNESS OF FIT TEST.///18HOESTIMACVM  650
     1TED BETA = ,1PD15.7/22HONUMBER OF FAILURES = ,I6)                 CVM  660
   90 FORMAT (32HOCRAMER - VON MISES STATISTIC = ,1PD15.7/52HOREJECT THECVM  670
     1 WEIBULL INTENSITY MODEL IF THE STATISTIC/48H EXCEEDS THE APPROPRICVM  680
     2ATE CRITICAL VALUE FOR M = ,I5/)                                  CVM  690
  100 FORMAT (77HOTHE CRAMER - VON MISES GOODNESS OF FIT TEST IS NOT APPCVM  700
     1ROPPIATE FOR THIS CASE/58H BECAUSE ONE OR MORE SYSTEMS HAVE NON-ZECVM  710
     2RO STARTING TIMES./)                                              CVM  720
  110 FORMAT (29HOUNBIASED ESTIMATE OF BETA = ,1PD15.7////)             CVM  730
      END                                                               CVM  740
```

```
      SUBROUTINE INDATA (X,C,NOGOOD,GOF,HALT,NFAIL,NSYS,M,K,IUNIT,ECHO) INDAA 10
COMMENT       THIS SUBROUTINE READS IN THE BEGINNING AND ENDING TIMES,  INDAA 20
C          AND THE FAILURE TIMES.                                       INDAA 30
C                                                                       INDAA 40
C          THE ARRANGEMENT OF INPUT IS AS FOLLOWS:                      INDAA 50
C                                                                       INDAA 60
C               BEGINNING AND ENDING TIMES FOR FIRST SYSTEM, FAILURE    INDAA 70
C                    TIMES FOR FIRST SYSTEM, NEGATIVE VALUE AS TRAILER. INDAA 80
C               BEGINNING AND ENDING TIMES FOR SECOND SYSTEM, FAILURE   INDAA 90
C                    TIMES FOR SECOND SYSTEM, NEGATIVE VALUE FOR TRAILER. INDAA100
C                                                                       INDAA110
C               ...                                                     INDAA120
C               ...                                                     INDAA130
C               ...                                                     INDAA140
C               BEGINNING AND ENDING TIMES FOR K-TH SYSTEM, FAILURE     INDAA150
C                    TIMES FOR K-TH SYSTEM, NEGATIVE VALUE AS TRAILER.  INDAA160
C               NEGATIVE VALUE TO MARK END OF CASE.                     INDAA170
C                    (REPEAT ABOVE FOR AS MANY CASES AS NEEDED.)        INDAA180
C               NEGATIVE VALUE TO MARK END OF RUN.                      INDAA190
C                                                                       INDAA200
C          INPUT IS FREE-FIELD, REQUIRING ONLY THAT AT LEAST ONE BLANK  INDAA210
C          COLUMN SEPARATE ADJACENT VALUES.                             INDAA220
C                                                                       INDAA230
      DOUBLE PRECISION X(NFAIL), C(NSYS,2)                              INDAA240
      LOGICAL ECHO, GOF, HALT, NOGOOD                                   INDAA250
C                                                                       INDAA260
      GOF=.TRUE.                                                        INDAA270
      HALT=.FALSE.                                                      INDAA280
      NOGOOD=.FALSE.                                                    INDAA290
      WRITE (6,130)                                                     INDAA300
C                                                                       INDAA310
      J=1                                                               INDAA320
      I=1                                                               INDAA330
C - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - INDAA340
C                    BEGIN INPUT CYCLE.                                 INDAA350
C - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - INDAA360
   10 CALL INREAL (IUNIT,C(J,1))                                        INDAA370
C                    NEGATIVE VALUE TO MARK THE END OF THIS CASE.       INDAA380
      IF (C(J,1).LT.0.0D0) GO TO 50                                     INDAA390
      CALL INREAL (IUNIT,C(J,2))                                        INDAA400
      IF (DABS(C(J,1)).GT.1.0D-08) GOF=.FALSE.                          INDAA410
      J=J+1                                                             INDAA420
      IF (J.LE.NSYS) GO TO 20                                           INDAA430
      WRITE (6,100) J                                                   INDAA440
      NOGOOD=.TRUE.                                                     INDAA450
C                                                                       INDAA460
   20 CALL INREAL (IUNIT,X(I))                                          INDAA470
C                    NEGATIVE VALUE TO MARK THE END OF THIS SYSTEM.     INDAA480
      IF (X(I).LT.0.0D0) GO TO 10                                       INDAA490
      IF (X(I).GT.1.0D-15) GO TO 30                                     INDAA500
      WRITE (6,110)                                                     INDAA510
      NOGOOD=.TRUE.                                                     INDAA520
C                                                                       INDAA530
C                    EACH FAILURE MUST FALL WITHIN THE TEST PERIOD.     INDAA540
   30 IF (C(J-1,1).LE.X(I).AND.X(I).LE.C(J-1,2)) GO TO 40               INDAA550
      WRITE (6,120) X(I),C(J-1,1),C(J-1,2)                              INDAA560
      NOGOOD=.TRUE.                                                     INDAA570
   40 IF (MOD(J,2).EQ.0) X(I)=-X(I)
```

```
      I=I+1                                                              INDAA580
      IF (I.LE.NFAIL) GO TO 20                                           INDAA590
      WRITE (6,99) I                                                     INDAA600
      NOGOOD=.TRUE.                                                      INDAA610
      GO TO 20                                                           INDAA620
C - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - INDAA630
C                  END OF INPUT CYCLE.                                   INDAA640
C - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - INDAA650
   50 M=I-1                                                              INDAA660
      IF (M.LE.0) GO TO 60                                               INDAA670
C                  TOTAL NUMBER OF FAILURES.                             INDAA680
      WRITE (6,70) M                                                     INDAA690
      K=J-1                                                              INDAA700
C                  TOTAL NUMBER OF SYSTEMS.                              INDAA710
      WRITE (6,80) K                                                     INDAA720
      IF (ECHO) CALL REPEAT (X,C,NFAIL,NSYS,M,K)                         INDAA730
      RETURN                                                             INDAA740
C - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - INDAA750
C                  END OF RUN.                                           INDAA760
C - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - INDAA770
   60 HALT=.TRUE.                                                        INDAA780
      WRITE (6,140)                                                      INDAA790
      RETURN                                                             INDAA800
C                                                                        INDAA810
   70 FORMAT (28HCTOTAL NUMBER OF FAILURES = ,I5)                        INDAA820
   80 FORMAT (28HCTOTAL NUMBER OF SYSTEMS  = ,I5)                        INDAA830
   90 FORMAT (18H0 INDATA ERROR -- ,I6,22H IS TOO MANY FAILURES./)       INDAA840
  100 FORMAT (18H0 INDATA ERROR -- ,I6,21H IS TOO MANY SYSTEMS./)        INDAA850
  110 FORMAT (52H0 INDATA ERROR -- A FAILURE AT 0.0000000 WAS INPUT. /71INDAA860
     1H0 THE PROBABILITY OF SUCH A FAILURE TIME IS 0.0 ACCORDING TO THE INDAA870
     2MODEL./)                                                           INDAA880
  120 FORMAT (33H0 INDATA ERROR -- THE FAILURE AT ,1PD10.3/44H  DOES NOTINDAA890
     1 FALL WITHIN THE TEST PERIOD FROM ,1PD10.3,4H TO ,1PD10.3/)        INDAA900
  130 FORMAT (19H1 DATA INPUT PHASE.//)                                  INDAA910
  140 FORMAT (28H0 PROGRAM RUN ENDS NORMALLY./1H1)                       INDAA920
      END                                                                INDAA930
```

```
      SUBROUTINE LOCIP (X,NFAIL,M,IP)                                   LOCIP 10
CUMMENT       THIS SUBROUTINE LOCATES THE POSITION OF THE LAST FAILURE  LOCIP 20
C         ASSOCIATED WITH THE SYSTEM WHICH HAD FAILURE 'X(IP)'.  ON     LOCIP 30
C         RETURN, 'IP' INDEXES THIS LAST FAILURE.                       LOCIP 40
C                                                                       LOCIP 50
      DOUBLE PRECISION X(NFAIL)                                         LOCIP 60
C                                                                       LOCIP 70
      IB=IP+1                                                           LOCIP 80
      MM1=M-1                                                           LOCIP 90
      DO 10 I=IB,MM1                                                    LOCIP100
      IF ((X(I).LT.0.0D0.AND.X(I+1).LT.0.0D0).OR.(X(I).GE.0.0D0.AND.X(I+LOCIP110
     11).GE.0.0D0)) GO TO 10                                            LOCIP120
      IP=I                                                              LOCIP130
      RETURN                                                            LOCIP140
   10 CONTINUE                                                          LOCIP150
      IP=M                                                              LOCIP160
      RETURN                                                            LOCIP170
      END                                                               LOCIP180
```

```
      SUBROUTINE LOGS (X,C,CLN,NFAIL,NSYS,M,K,SUMLNX)                   LOGS  10
CUMMENT       THIS SUBROUTINE TAKES LOGARITHMS.                         LOGS  20
C                                                                       LOGS  30
      DOUBLE PRECISION X(NFAIL), C(NSYS,2), CLN(NSYS,2)                 LOGS  40
      DOUBLE PRECISION SUMLNX                                           LOGS  50
C                                                                       LOGS  60
C                    LOGARITHMS OF BEGINNING AND ENDING TIMES.          LOGS  70
      DO 20 I=1,K                                                       LOGS  80
      CLN(I,2)=DLOG(C(I,2))                                             LOGS  90
      IF (C(I,1).LE.0.0D0) GO TO 10                                     LOGS 100
      CLN(I,1)=DLOG(C(I,1))                                             LOGS 110
      GO TO 20                                                          LOGS 120
   10 C(I,1)=0.0D0                                                      LOGS 130
      CLN(I,1)=0.0D0                                                    LOGS 140
   20 CONTINUE                                                          LOGS 150
C                                                                       LOGS 160
C                    SUM OF LOGARITHMS OF FAILURES.                     LOGS 170
      SUMLNX=0.0D0                                                      LOGS 180
      DO 30 I=1,M                                                       LOGS 190
      SUMLNX=SUMLNX+DLOG(DABS(X(I)))                                    LOGS 200
   30 CONTINUE                                                          LOGS 210
      RETURN                                                            LOGS 220
C                                                                       LOGS 230
      END                                                               LOGS 240
```

```
      SUBROUTINE MLEW (SUMLNX,C,CB,CLN,NOGOOD,NSYS,M,K,BETA)              MLEW  10
C-                                                                        MLEW  20
C-                                                                        MLEW  30
C-       ESTIMATES OF PARAMETERS OF THE WEIBULL INTENSITY MODEL.          MLEW  40
C-                                                                        MLEW  50
C-              THIS SUBROUTINE ESTIMATES BETA AND LAMBDA OF THE          MLEW  60
C-     WEIBULL INTENSITY FUNCTION  R(X)=LAMBDA*BETA*X**(BETA-1.0)        MLEW  70
C-              BETA IS DETERMINED BY AN ITERATIVE PROCESS WHICH          MLEW  80
C-     EXAMINES THE SIGNED DIFFERENCE OF A CONSTANT MINUS A FUNCTION OF   MLEW  90
C-     BETA AS BETA IS DECREASED FROM A LARGE INITIAL ESTIMATE BY         MLEW 100
C-     NON-POSITIVE POWERS OF 10.0 UNTIL AN EPSILON TOLERANCE IS          MLEW 110
C-     SATISFIED OR BETA IS LESS THAN 0.00000001 IN VALUE.                MLEW 120
C-              LAMBDA IS CALCULATED BASED ON BETA.                       MLEW 130
C                                                                         MLEW 140
      DOUBLE PRECISION C(NSYS,2), CB(NSYS,2), CLN(NSYS,2)                 MLEW 150
      DOUBLE PRECISION BETA, LAMBDA, EPSILN                               MLEW 160
      DOUBLE PRECISION DENOM, DENOM1, DENOM2, TOP, TOP1, TOP2             MLEW 170
      DOUBLE PRECISION A, ABDI, ADJ, D, DIFF, TOTFAL                      MLEW 180
      DOUBLE PRECISION SUMLNX                                             MLEW 190
      LOGICAL NOGOOD                                                      MLEW 200
C                                                                         MLEW 210
      WRITE (6,110)                                                       MLEW 220
      BETA=1.0D+01                                                        MLEW 230
      EPSILN=1.0D-05                                                      MLEW 240
      TOTFAL=DBLE(FLOAT(M))                                               MLEW 250
      NPDIFF=0                                                            MLEW 260
      NOGOOD=.FALSE.                                                      MLEW 270
C                                                                         MLEW 280
C                  CONSTANT NOT INVOLVING BETA.                           MLEW 290
      A=SUMLNX/TOTFAL                                                     MLEW 300
      WRITE (6,130) A                                                     MLEW 310
C - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -   MLEW 320
C                  BEGIN ITERATION PROCEDURE.                             MLEW 330
C - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -   MLEW 340
      ADJ=1.0D0                                                           MLEW 350
      IZERO=0                                                             MLEW 360
      WRITE (6,150)                                                       MLEW 370
C                                                                         MLEW 380
   10 TOP1=0.0D0                                                          MLEW 390
      TOP2=0.0D0                                                          MLEW 400
      DENOM1=0.0D0                                                        MLEW 410
      DENOM2=0.0D0                                                        MLEW 420
      DO 20 I=1,K                                                         MLEW 430
      DO 20 J=1,2                                                         MLEW 440
      CB(I,J)=C(I,J)**BETA                                                MLEW 450
   20 CONTINUE                                                            MLEW 460
C                                                                         MLEW 470
      DO 30 I=1,K                                                         MLEW 480
      TOP1=TOP1+CB(I,1)*CLN(I,1)                                          MLEW 490
      TOP2=TOP2+CB(I,2)*CLN(I,2)                                          MLEW 500
      DENOM1=DENOM1+CB(I,1)                                               MLEW 510
      DENOM2=DENOM2+CB(I,2)                                               MLEW 520
   30 CONTINUE                                                            MLEW 530
C                                                                         MLEW 540
      TOP=TOP2-TOP1                                                       MLEW 550
      DENOM=DENOM2-DENOM1                                                 MLEW 560
      D=(TOP/DENOM)-1.0D0/BETA                                            MLEW 570
```

```
      DIFF=A-D                                                          MLEW 580
      LAMBDA=TOTFAL/DENOM                                               MLEW 590
      WRITE (6,120) BETA,D,DIFF,LAMBDA                                  MLEW 600
      ABDI=DABS(DIFF)                                                   MLEW 610
      IF (ABDI.LE.EPSILN) GO TO 50                                      MLEW 620
      IF (DIFF.GT.0.0D0) GO TO 40                                       MLEW 630
      NPDIFF=0                                                          MLEW 640
      BETA=BETA-ADJ                                                     MLEW 650
      IF (BETA.LE.1.0D-15) GO TO 60                                     MLEW 660
      GO TO 10                                                          MLEW 670
C - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - MLEW 680
C        BETA TOO SMALL -- DECREASE STEP SIZE AND USE PREVIOUS BETA.    MLEW 690
C - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - MLEW 700
   40 BETA=BETA+ADJ                                                     MLEW 710
      NPDIFF=NPDIFF+1                                                   MLEW 720
      IF (NPDIFF.GT.10) GO TO 70                                        MLEW 730
      ADJ=1.0D-1*ADJ                                                    MLEW 740
      IF (DABS(DLOG10(DABS(BETA))-DLOG10(ADJ)).GT.1.5D+1) GO TO 80      MLEW 750
      BETA=BETA-ADJ                                                     MLEW 760
      GO TO 10                                                          MLEW 770
C - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - MLEW 780
C                  EPSILON TOLERANCE MET.                               MLEW 790
C - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - MLEW 800
   50 WRITE (6,140) BETA,LAMBDA,ABDI,EPSILN,ADJ                         MLEW 810
      GO TO 100                                                         MLEW 820
C                                                                       MLEW 830
   60 IZERO=IZERO+1                                                     MLEW 840
      IF (IZERO.LE.8) GO TO 40                                          MLEW 850
      WRITE (6,160)                                                     MLEW 860
      GO TO 90                                                          MLEW 870
   70 WRITE (6,170)                                                     MLEW 880
      GO TO 90                                                          MLEW 890
   80 WRITE (6,180)                                                     MLEW 900
   90 NOGOOD=.TRUE.                                                     MLEW 910
C                                                                       MLEW 920
  100 RETURN                                                            MLEW 930
C                                                                       MLEW 940
C                                                                       MLEW 950
  110 FORMAT (65H1 ESTIMATION OF THE PARAMETERS OF THE WEIBULL INTENSITYMLEW 960
     1 FUNCTION,/)                                                      MLEW 970
  120 FORMAT (1H ,4(1PD16.9,4X))                                        MLEW 980
  130 FORMAT (35HOCONSTANT NOT INVOLVING BETA:  A = ,1PD15.7/)          MLEW 990
  140 FORMAT (30H1THE FINAL ESTIMATE OF BETA = ,1PD15.7//32H THE FINAL EMLEW1000
     1STIMATE OF LAMBDA = ,1PD15.7//16H CONVERGENCE TO ,1PD15.7/30H WHICMLEW1010
     2H IS LESS THAN EPSILON = ,1PD15.7//24H THE FINAL STEP SIZE IS ,1PDMLEW1020
     315.7/)                                                            MLEW1030
  150 FORMAT (1HO/16HO ESTIMATED BETA,5X,15H FUNCTION D(B'),8X,10H A - DMLEW1040
     1(B'),6X,17H ESTIMATED LAMBDA/)                                    MLEW1050
  160 FORMAT (41HO MLEW ERROR -- BETA LESS THAN 0.00000001)             MLEW1060
  170 FORMAT (53HO MLEW ERROR -- INITIAL ESTIMATE OF BETA IS TOO SMALL) MLEW1070
  180 FORMAT (50HO MLEW ERROR -- STEP-SIZE HAS BECOME INSIGNIFICANT/2OH MLEW1080
     1BETA NOT RESOLVABLE/)                                             MLEW1090
      END                                                               MLEW1100
```

```
      SUBROUTINE PRNTX (X,NFAIL,IB,IE)                                  PRNTX 10
COMMENT       THIS SUBROUTINE PRINTS THE FAILURES 'X(IB)' THROUGH       PRNTX 20
C          'X(IE)'.                                                     PRNTX 30
C                                                                       PRNTX 40
      DOUBLE PRECISION X(NFAIL)                                         PRNTX 50
      DOUBLE PRECISION XTEMP(5)                                         PRNTX 60
      DATA L /5/                                                        PRNTX 70
C                                                                       PRNTX 80
      IT=0                                                              PRNTX 90
      DO 20 I=IB,IE                                                     PRNTX100
      IT=IT+1                                                           PRNTX110
      XTEMP(IT)=DABS(X(I))                                              PRNTX120
      IF (I.EQ.IE) GO TO 10                                             PRNTX130
      IF (IT.LT.L) GO TO 20                                             PRNTX140
   10 WRITE (6,30) (XTEMP(J),J=1,IT)                                    PRNTX150
      IT=0                                                              PRNTX160
   20 CONTINUE                                                          PRNTX170
C                                                                       PRNTX180
      RETURN                                                            PRNTX190
C                                                                       PRNTX200
   30 FORMAT (1H ,30X,1P5D10.3)                                         PRNTX210
      END                                                               PRNTX220
```

```
      SUBROUTINE REPEAT (X,C,NFAIL,NSYS,M,K)                            REPET 10
COMMENT       THIS SUBROUTINE REPEATS THE INPUT DATA.                   REPET 20
C                                                                       REPET 30
      DOUBLE PRECISION X(NFAIL), C(NSYS,2)                              REPET 40
C                                                                       REPET 50
      WRITE (6,20)                                                      REPET 60
      IP=0                                                              REPET 70
C                                                                       REPET 80
      DO 10 J=1,K                                                       REPET 90
      WRITE (6,30) C(J,1),C(J,2)                                        REPET100
      IB=IP+1                                                           REPET110
      CALL LOCIP (X,NFAIL,M,IP)                                         REPET120
      IE=IP                                                             REPET130
      CALL PRNTX (X,NFAIL,IB,IE)                                        REPET140
   10 CONTINUE                                                          REPET150
C                                                                       REPET160
      RETURN                                                            REPET170
C                                                                       REPET180
   20 FORMAT (1H0/34HOSYSTEM STARTING AND ENDING TIMES./1H0,35X,10H FAILREPET190
     1URES./)                                                           REPET200
   30 FORMAT (1H ,1P2D12.3)                                             REPET210
      END                                                               REPET220
```

```
      SUBROUTINE SORTX (X,N)                                                SORTX 10
COMMENT       THIS SUBROUTINE SORTS THE VECTOR X INTO INCREASING ORDER.     SORTX 20
C                                                                           SORTX 30
      DOUBLE PRECISION X(N)                                                 SORTX 40
      M=N                                                                   SORTX 50
   10 M=M/2                                                                 SORTX 60
      IF (M.EQ.0) RETURN                                                    SORTX 70
      K=N-M+1                                                               SORTX 80
      J=1                                                                   SORTX 90
   20 I=J                                                                   SORTX100
   30 L=I+M                                                                 SORTX110
      IF (X(I).GT.X(L)) GO TO 50                                            SORTX120
   40 J=J+1                                                                 SORTX130
      IF (J-K) 20,10,10                                                     SORTX140
   50 T=X(L)                                                                SORTX150
      X(L)=X(I)                                                             SORTX160
      X(I)=T                                                                SORTX170
      I=I-M                                                                 SORTX180
      IF (I) 40,40,30                                                       SORTX190
      END                                                                   SORTX200
```

```
      SUBROUTINE TRANSF (X,TQ,IB,IBT,IET,NFAIL)                    TRANF 10
COMMENT       THIS SUBROUTINE TRANSFORMS THE FAILURE TIMES.        TRANF 20
C                                                                  TRANF 30
      DOUBLE PRECISION X(NFAIL), TQ                                TRANF 40
C                                                                  TRANF 50
      IF (IET.LE.0) RETURN                                         TRANF 60
      J=IB                                                         TRANF 70
      DO 10 I=IBT,IET                                              TRANF 80
      X(I)=X(J)/TQ                                                 TRANF 90
      J=J+1                                                        TRANF100
   10 CONTINUE                                                     TRANF110
      RETURN                                                       TRANF120
      END                                                          TRANF130
```

```
      SUBROUTINE INREAL (CHANNL,DESTIN)                                  INREL 10
COMMENT       FREE FIELD READ.                                           INREL 20
C         (A FORTRAN TRANSLATION OF ACM ALGORITHM 239.)                  INREL 30
C                                                                        INREL 40
C             EACH CALL OF THIS SUBROUTINE WILL READ ONE REAL NUMBER     INREL 50
C         FROM UNIT 'CHANNL', CONVERT IT, AND STORE IT IN 'DESTIN'.      INREL 60
C                                                                        INREL 70
      INTEGER CHANNL                                                     INREL 80
      DOUBLE PRECISION DESTIN                                            INREL 90
      REAL SIG, FP, D                                                    INREL100
      INTEGER ESIG, EP, IP, CH                                           INREL110
      INTEGER CHAR, UNSGND                                               INREL120
C                                                                        INREL130
      SIG=1.0                                                            INREL140
      EP=0                                                               INREL150
      FP=0                                                               INREL160
C                                                                        INREL170
   10 CH=CHAR(CHANNL)                                                    INREL180
C        SUPPRESS INITIAL BLANKS.                                        INREL190
      IF (CH.EQ.14) GO TO 10                                             INREL200
C        12 = '+' AND 11 = '-'.                                          INREL210
      IF (CH.NE.12) GO TO 20                                             INREL220
      CH=CHAR(CHANNL)                                                    INREL230
      GO TO 30                                                           INREL240
   20 IF (CH.NE.11) GO TO 30                                             INREL250
      SIG=-1.0                                                           INREL260
      CH=CHAR(CHANNL)                                                    INREL270
   30 CONTINUE                                                           INREL280
      IF (CH.GT.10) GO TO 70                                             INREL290
      IF (CH.GE.10) GO TO 40                                             INREL300
      IP=UNSGND(CHANNL,CH)                                               INREL310
      GO TO 50                                                           INREL320
   40 CONTINUE                                                           INREL330
      IP=0                                                               INREL340
   50 CONTINUE                                                           INREL350
      IF (CH.NE.10) GO TO 100                                            INREL360
      CH=CHAR(CHANNL)                                                    INREL370
      FP=0                                                               INREL380
      IF (CH.GE.10) GO TO 100                                            INREL390
      D=0.1                                                              INREL400
C                                                                        INREL410
   60 FP=FP+FLOAT(CH)*D                                                  INREL420
      D=D*0.1                                                            INREL430
      CH=CHAR(CHANNL)                                                    INREL440
      IF (CH.LT.10) GO TO 60                                             INREL450
      GO TO 100                                                          INREL460
   70 CONTINUE                                                           INREL470
      IF (CH.NE.13) GO TO 80                                             INREL480
      IP=1                                                               INREL490
      GO TO 90                                                           INREL500
   80 CONTINUE                                                           INREL510
      WRITE (6,180)                                                      INREL520
      STOP                                                               INREL530
   90 CONTINUE                                                           INREL540
  100 CONTINUE                                                           INREL550
C                                                                        INREL560
      IF (CH.NE.13) GO TO 160                                            INREL570
```

```
      CH=CHAR(CHANNL)                                                   INREL580
      ESIG=1                                                            INREL590
      IF (CH.NE.12.AND.CH.NE.14) GO TO 110                              INREL600
      CH=CHAR(CHANNL)                                                   INREL610
      GO TO 130                                                         INREL620
  110 CONTINUE                                                          INREL630
      IF (CH.NE.11) GO TO 120                                           INREL640
C                     NEGATIVE EXPONENT.                                INREL650
      ESIG=-1                                                           INREL660
      CH=CHAR(CHANNL)                                                   INREL670
  120 CONTINUE                                                          INREL680
  130 CONTINUE                                                          INREL690
      IF (CH.GE.10) GO TO 140                                           INREL700
      EP=UNSGND(CHANNL,CH)*ESIG                                         INREL710
      GO TO 150                                                         INREL720
  140 CONTINUE                                                          INREL730
      WRITE (6,190)                                                     INREL740
      STOP                                                              INREL750
  150 CONTINUE                                                          INREL760
  160 CONTINUE                                                          INREL770
C                                                                       INREL780
      IF (CH.NE.14) GO TO 170                                           INREL790
C                                                                       INREL800
      DESTIN=DBLE(SIG*(FLOAT(IP)+FP)*(10.0**EP))                        INREL810
C                                                                       INREL820
      RETURN                                                            INREL830
  170 WRITE (6,200)                                                     INREL840
      STOP                                                              INREL850
C                                                                       INREL860
  180 FORMAT (36H0 INREAL ERROR -- CH OUT OF RANGE.  /)                 INREL870
  190 FORMAT (41H0 INREAL ERROR -- EXPONENT NOT DIGIT.    /)            INREL880
  200 FORMAT (53H0 INREAL ERROR -- NO BLANK FOUND BETWEEN DATA VALUES./)INREL890
      END                                                               INREL900
```

```
      INTEGER FUNCTION CHAR (CHANNL)                                    CHAR  10
COMMENT       'CHAR' RETURNS AN INTEGER VALUE FOR THE NEXT CHARACTER    CHAR  20
C         ON UNIT 'CHANNL'.                                             CHAR  30
C                                                                       CHAR  40
      INTEGER C                                                         CHAR  50
      INTEGER CHANNL                                                    CHAR  60
      INTEGER STRING(15)                                                CHAR  70
      DATA STRING(1), STRING(2), STRING(3), STRING(4), STRING(5), STRINGCHAR  80
     1(6), STRING(7), STRING(8), STRING(9), STRING(10), STRING(11), STRICHAR  90
     2NG(12), STRING(13), STRING(14), STRING(15) /1H0, 1H1, 1H2, 1H3, 1HCHAR 100
     34, 1H5, 1H6, 1H7, 1H8, 1H9, 1H., 1H-, 1H+, 1HE, 1H /              CHAR 110
      DATA LSTR /15/                                                    CHAR 120
      CALL INSYML (CHANNL,STRING,LSTR,C)                                CHAR 130
C         IS CHARACTER LEGAL?                                           CHAR 140
      IF (C.LE.0) GO TO 10                                              CHAR 150
      CHAR=C-1                                                          CHAR 160
      RETURN                                                            CHAR 170
   10 WRITE (6,20)                                                      CHAR 180
      STOP                                                              CHAR 190
C                                                                       CHAR 200
   20 FORMAT (58H0 CHAR ERROR -- ILLEGAL INPUT CHARACTER.  PROGRAM ABORTCHAR 210
     1ED./)                                                             CHAR 220
      END                                                               CHAR 230
```

```
      SUBROUTINE INSYML (IUNIT,STRING,LSTR,I)                          INSYL 10
COMMENT       THIS SUBROUTINE EXAMINES THE NEXT CHARACTER ON 'IUNIT'    INSYL 20
C        AND DETERMINES ITS POSITION NUMBER 'I' WITHIN THE 'STRING' OF  INSYL 30
C        LENGTH 'LSTR'.                                                 INSYL 40
C                                                                       INSYL 50
      INTEGER STRING(LSTR)                                             INSYL 60
      INTEGER BUFFER(160)                                              INSYL 70
      DATA IP /0/                                                      INSYL 80
      DATA LENGTH /80/                                                 INSYL 90
C                                                                      INSYL100
      IF (IP.NE.0) GO TO 10                                            INSYL110
C            FILL INPUT BUFFER.                                        INSYL120
      READ (IUNIT,20) (BUFFER(J),J=1,LENGTH)                           INSYL130
   10 CONTINUE                                                         INSYL140
      IP=IP+1                                                          INSYL150
      IC=IP                                                            INSYL160
C        MATCH THE CHARACTER.                                          INSYL170
      CALL MATCH (BUFFER(IC),STRING,LSTR,I)                            INSYL180
C        IF POINTER 'IP' HAS REACHED THE END OF A LINE, RESET IT.      INSYL190
      IF (IP.EQ.LENGTH) IP=0                                           INSYL200
C                                                                      INSYL210
      RETURN                                                           INSYL220
C                                                                      INSYL230
   20 FORMAT (128A1)                                                   INSYL240
      END                                                              INSYL250
```

```
      SUBROUTINE MATCH (CHAR,STRING,LSTR,IP)                            MATCH 10
COMMENT       THIS SUBROUTINE FINDS THE POSITION 'IP' OF 'CHAR' IN      MATCH 20
C         'STRING' WHICH HAS A LENGTH OF 'LSTR'.                        MATCH 30
C                                                                       MATCH 40
      INTEGER CHAR, STRING(LSTR)                                        MATCH 50
      IP=0                                                              MATCH 60
      DO 10 I=1,LSTR                                                    MATCH 70
      IF (STRING(I).NE.CHAR) GO TO 10                                   MATCH 80
      IP=I                                                              MATCH 90
      GO TO 20                                                          MATCH100
   10 CONTINUE                                                          MATCH110
      WRITE (6,30) CHAR                                                 MATCH120
   20 RETURN                                                            MATCH130
C                                                                       MATCH140
   30 FORMAT (31HO MATCH ERROR -- THE CHARACTER ,A1,16H IS NOT MATCHED./MATCH150
     1)                                                                 MATCH160
      END                                                               MATCH170
```

```
      INTEGER FUNCTION UNSGND (CHANNL,CH)                                UNSG  10
COMMENT       THIS FUNCTION RETURNS THE NEXT UNSIGNED INTEGER FROM       UNSG  20
C         'CHANNL'.                                                      UNSG  30
C                                                                        UNSG  40
      INTEGER CHANNL, CH                                                 UNSG  50
      INTEGER CHAR                                                       UNSG  60
      INTEGER U                                                          UNSG  70
      U=0                                                                UNSG  80
   10 U=10*U+CH                                                          UNSG  90
      CH=CHAR(CHANNL)                                                    UNSG 100
      IF (CH.LT.10) GO TO 10                                             UNSG 110
      UNSGND=U                                                           UNSG 120
      RETURN                                                             UNSG 130
      END                                                                UNSG 140
```

Next page is blank.

APPENDIX B

INPUT FOR TEST CASES

Next page is blank.

INPUT DATA FOR TEST CASES

```
  1.0  100.0  32.  44.  56.  75.  95.  -99.  -99.
12.2  45.3  10.0  14.5  16.8  457.9  -99 -99
  0.0  3256.3  .7  3.7  13.2  17.6  54.5  99.2  112.2  120.9  151.0  163.0
 174.5  191.6  282.8
 355.2  486.3  490.5  513.3  558.4  678.1  688.0  785.9  887.0  1010.7
  1029.1  1034.4  1136.1  1178.9  1259.7  1297.9  1419.7  1571.7
1629.8  1702.3  1928.9  2072.3  2525.2  2928.5  3016.4  3181.0
 3256.3
 -99.  -99.
  0.0  79.0  0.0  20  45.  62  64  68  76  -8 -7
  456.  789.  456  456.1  457  458  489  -99  -99
  0.0  197.2  4.3  4.4  10.2  23.5  23.8  26.4  74.0  77.1  92.1  197.2
-999.
  0.0  190.8  0.1  5.6  18.6  19.5  24.2  26.7  45.1  45.8  75.7  79.7  98.6
              120.1  161.8  180.6  190.8
  -999
  0.0  195.8  8.4  32.5  44.7  48.4  50.6  73.6  98.7  112.2  129.8  136.0
              195.8
  -999
  -999
0.0  800.0  45.  456.  467.  477.  484.  492.
 0.12  564.0  65.0 78. 89. 99.1 -99 -99
  -99
```


Next page is blank.

APPENDIX C

OUTPUT FOR TEST CASES

Next page is blank.

WEIBULL INTENSITY MODEL PARAMETER ESTIMATION AND GOODNESS OF FIT TEST

VERSION OF 27 JULY 1979

DATA INPUT PHASE.

TOTAL NUMBER OF FAILURES = 5

TOTAL NUMBER OF SYSTEMS = 1

SYSTEM STARTING AND ENDING TIMES.

FAILURES.

1.000D+00 1.000D+02

3.200D+01 4.400D+01 5.600D+01 7.500D+01 9.500D+01

ESTIMATION OF THE PARAMETERS OF THE WEIBULL INTENSITY FUNCTION.

CONSTANT NOT INVOLVING BETA: A = 4.0293284D+00

ESTIMATED BETA	FUNCTION D(B')	A - D(B')	ESTIMATED LAMBDA
1.000000000D+01	4.505170186D+00	-4.758417395D-01	5.000000000D-20
9.000000000D+00	4.494059075D+00	-4.647306284D-01	5.000000000D-18
8.000000000D+00	4.480170186D+00	-4.508417395D-01	5.000000000D-16
7.000000000D+00	4.462313043D+00	-4.329845966D-01	5.000000000D-14
6.000000000D+00	4.438503519D+00	-4.091750728D-01	5.000000000D-12
5.000000000D+00	4.405170186D+00	-3.758417399D-01	5.000000001D-10
4.000000000D+00	4.355170232D+00	-3.258417855D-01	5.000000050D-08
3.000000000D+00	4.271841458D+00	-2.425130113D-01	5.000005000D-06
2.000000000D+00	4.105630749D+00	-7.630230255D-02	5.000500050D-04
1.000000000D+00	3.651687057D+00	3.776413900D-01	5.050505051D-02
1.900000000D+00	4.079584382D+00	-5.025593598D-02	7.925722105D-04
1.800000000D+00	4.050771688D+00	-2.144324100D-02	1.256258774D-03
1.700000000D+00	4.018768973D+00	1.055947321D-02	1.991328615D-03
1.790000000D+00	4.047722570D+00	-1.839412345D-02	1.315480002D-03
1.780000000D+00	4.044641152D+00	-1.531270529D-02	1.377493745D-03
1.770000000D+00	4.041526963D+00	-1.219851680D-02	1.442431753D-03
1.760000000D+00	4.038379527D+00	-9.051080574D-03	1.510432003D-03
1.750000000D+00	4.035198358D+00	-5.869911384D-03	1.581638988D-03
1.740000000D+00	4.031982963D+00	-2.654516074D-03	1.656204028D-03
1.730000000D+00	4.028732840D+00	5.956065495D-04	1.734285593D-03
1.739000000D+00	4.031659521D+00	-2.331074881D-03	1.663851262D-03
1.738000000D+00	4.031335732D+00	-2.007265916D-03	1.671533818D-03
1.737000000D+00	4.031011595D+00	-1.683148672D-03	1.679251859D-03
1.736000000D+00	4.030687109D+00	-1.358662642D-03	1.687005549D-03
1.735000000D+00	4.030362274D+00	-1.033827318D-03	1.694795052D-03
1.734000000D+00	4.030037089D+00	-7.086421920D-04	1.702620534D-03
1.733000000D+00	4.029711553D+00	-3.831067529D-04	1.710482162D-03
1.732000000D+00	4.029385667D+00	-5.722049074D-05	1.718380103D-03
1.731000000D+00	4.029059429D+00	2.690171060D-04	1.726314523D-03
1.731900000D+00	4.029353059D+00	-2.461254957D-05	1.719171900D-03
1.731800000D+00	4.029320448D+00	7.998054560D-06	1.719964063D-03

THE FINAL ESTIMATE OF BETA = 1.7318000D+00

THE FINAL ESTIMATE OF LAMBDA = 1.7199641D-03

CONVERGENCE TO 7.9989355D-06
WHICH IS LESS THAN EPSILON = 1.0000000D-05

THE FINAL STEP SIZE IS 1.0000000D-04

CRAMER - VON MISES GOODNESS OF FIT TEST.

ESTIMATED BETA = 1.7318000D+00

NUMBER OF FAILURES = 5

UNBIASED ESTIMATE OF BETA = 1.3854400D+00

THE CRAMER - VON MISES GOODNESS OF FIT TEST IS NOT APPROPRIATE FOR THIS CASE BECAUSE ONE OR MORE SYSTEMS HAVE NON-ZERO STARTING TIMES.

```
 DATA INPUT PHASE.


 INDATA ERROR -- THE FAILURE AT  1.0000D+01
 DOES NOT FALL WITHIN THE TEST PERIOD FROM  1.2200D+01 TO  4.5300D+01

 INDATA ERROR -- THE FAILURE AT  4.5790D+02
 DOES NOT FALL WITHIN THE TEST PERIOD FROM  1.2200D+01 TO  4.5300D+01

TOTAL NUMBER OF FAILURES =      4

TOTAL NUMBER OF SYSTEMS  =      1


SYSTEM STARTING AND ENDING TIMES.

                                        FAILURES.

   1.2200D+01   4.5300D+01
                                1.0000D+01 1.4500D+01 1.6800D+01 4.5790D+02
```

DATA INPUT PHASE.

TOTAL NUMBER OF FAILURES = 40

TOTAL NUMBER OF SYSTEMS = 1

SYSTEM STARTING AND ENDING TIMES.

FAILURES.

0. 3.256D+03

7.000D-01 3.700D+00 1.320D+01 1.760D+01 5.450D+01
9.920D+01 1.122D+02 1.209D+02 1.510D+02 1.630D+02
1.745D+02 1.916D+02 2.828D+02 3.552D+02 4.863D+02
4.905D+02 5.133D+02 5.584D+02 6.781D+02 6.880D+02
7.859D+02 8.870D+02 1.011D+03 1.029D+03 1.034D+03
1.136D+03 1.179D+03 1.260D+03 1.298D+03 1.420D+03
1.572D+03 1.630D+03 1.702D+03 1.929D+03 2.072D+03
2.525D+03 2.929D+03 3.016D+03 3.181D+03 3.256D+03

ESTIMATION OF THE PARAMETERS OF THE WEIBULL INTENSITY FUNCTION.

CONSTANT NOT INVOLVING BETA: A = 6.0464988D+00

ESTIMATED BETA	FUNCTION D(B')	A - D(B')	ESTIMATED LAMBDA
1.000000000D+01	7.988346860D+00	-1.941848101D+00	2.98411736CD-34
9.000000000D+00	7.977235749D+00	-1.930736990D+00	9.717181358D-31
8.000000000D+00	7.963346860D+00	-1.916848101D+00	3.164205766D-27
7.000000000D+00	7.945489718D+00	-1.898990958D+00	1.030360323D-23
6.000000000D+00	7.921680194D+00	-1.875181434D+00	3.355162321D-20
5.000000000D+00	7.888346860D+00	-1.841848101D+00	1.092541507D-16
4.000000000D+00	7.838346860D+00	-1.791848101D+00	3.557642908D-13
3.000000000D+00	7.755013527D+00	-1.708514768D+00	1.158475260D-09
2.000000000D+00	7.588346860D+00	-1.541848101D+00	3.772342990D-06
1.000000000D+00	7.088346860D+00	-1.041848101D+00	1.228388048D-02
9.000000000D-01	6.977235749D+00	-9.307369898D-01	2.758087392D-02
8.000000000D-01	6.838346860D+00	-7.918481009D-01	6.192706024D-02
7.000000000D-01	6.659775432D+00	-6.132766723D-01	1.390442087D-01
6.000000000D-01	6.421680194D+00	-3.751814342D-01	3.121945705D-01
5.000000000D-01	6.088346860D+00	-4.184810091D-02	7.009673453D-01
4.000000000D-01	5.588346860D+00	4.581518991D-01	1.573874966D+00
4.900000000D-01	6.047530534D+00	-1.031774382D-03	7.600200173D-01
4.800000000D-01	6.005013527D+00	4.148523242D-02	8.240475545D-01
4.890000000D-01	6.043357085D+00	3.141674139D-03	7.661922507D-01
4.899000000D-01	6.047113956D+00	-6.151962382D-04	7.606349965D-01
4.898000000D-01	6.046697208D+00	-1.984479930D-04	7.612504734D-01
4.897000000D-01	6.046280289D+00	2.184704576D-04	7.618664482D-01
4.897900000D-01	6.046655523D+00	-1.567638089D-04	7.613120484D-01
4.897800000D-01	6.046613837D+00	-1.150779227D-04	7.613736285D-01
4.897700000D-01	6.046572150D+00	-7.339033417D-05	7.614352135D-01
4.897600000D-01	6.046530461D+00	-3.170104328D-05	7.614968035D-01
4.897500000D-01	6.046488770D+00	9.989950084D-06	7.615583985D-01

```
THE FINAL ESTIMATE OF BETA =    4.8975000D-01

THE FINAL ESTIMATE OF LAMBDA =    7.6155840D-01

CONVERGENCE TO   9.9899501D-06
WHICH IS LESS THAN EPSILON =    1.0000000D-05

THE FINAL STEP SIZE IS    1.0000000D-05
```

CRAMER - VON MISES GOODNESS OF FIT TEST.

ESTIMATED BETA = 4.8975000D-01

NUMBER OF FAILURES = 40

UNBIASED ESTIMATE OF BETA = 4.6526250D-01

CRAMER - VON MISES STATISTIC = 6.8281012D-02

REJECT THE WEIBULL INTENSITY MODEL IF THE STATISTIC
EXCEEDS THE APPROPRIATE CRITICAL VALUE FOR M = 39

DATA INPUT PHASE.

INDATA ERROR -- A FAILURE AT 0.0000000 WAS INPUT.

THE PROBABILITY OF SUCH A FAILURE TIME IS 0.0 ACCORDING TO THE MODEL.

TOTAL NUMBER OF FAILURES = 7

TOTAL NUMBER OF SYSTEMS = 1

SYSTEM STARTING AND ENDING TIMES.

		FAILURES.
0.	7.900D+01	0.

DATA INPUT PHASE.

TOTAL NUMBER OF FAILURES = 5

TOTAL NUMBER OF SYSTEMS = 1

SYSTEM STARTING AND ENDING TIMES.

FAILURES.

4.560D+02 7.890D+02

4.560D+02 4.561D+02 4.570D+02 4.580D+02 4.890D+02

ESTIMATION OF THE PARAMETERS OF THE WEIBULL INTENSITY FUNCTION.

CONSTANT NOT INVOLVING BETA: A = 6.1378240D+00

ESTIMATED BETA	FUNCTION D(B')	A - D(B')	ESTIMATED LAMBDA
1.0000000000D+01	6.5730555290D+00	-4.3523153700D-01	5.3704355370D-29
9.0000000000D+00	6.5636282560D+00	-4.2580426400D-01	4.2502328940D-26
8.0000000000D+00	6.5526773070D+00	-4.1485331580D-01	3.3712739540D-23
7.0000000000D+00	6.5399780970D+00	-4.0215410580D-01	2.6846472360D-20
6.0000000000D+00	6.5253232290D+00	-3.8749923780D-01	2.1527927750D-17
5.0000000000D+00	6.5085569710D+00	-3.7073297930D-01	1.7479656280D-14
4.0000000000D+00	6.4896198020D+00	-3.5179581000D-01	1.4522429250D-11
3.0000000000D+00	6.4685963840D+00	-3.3077239200D-01	1.2615104850D-08
2.0000000000D+00	6.4457540820D+00	-3.0793009020D-01	1.2060253020D-05
1.0000000000D+00	6.4215552730D+00	-2.8373128180D-01	1.5015015020D-02
9.0000000000D-01	6.4190838880D+00	-2.8125989690D-01	3.1704011970D-02
8.0000000000D-01	6.4166058560D+00	-2.7878186490D-01	6.7762626330D-02
7.0000000000D-01	6.4141218920D+00	-2.7629790050D-01	1.4709499980D-01
6.0000000000D-01	6.4116327180D+00	-2.7380872670D-01	3.2587604730D-01
5.0000000000D-01	6.4091390650D+00	-2.7131507380D-01	7.4239189660D-01
4.0000000000D-01	6.4066416700D+00	-2.6881767870D-01	1.7613042810D+00
3.0000000000D-01	6.4041412750D+00	-2.6631728330D-01	4.4561084710D+00
2.0000000000D-01	6.4016386250D+00	-2.6381463390D-01	1.2680050910D+01
1.0000000000D-01	6.3991344720D+00	-2.6131048020D-01	4.8096807790D+01
9.0000000000D-02	6.3988840030D+00	-2.6106601100D-01	5.6972370190D+01
8.0000000000D-02	6.3986335270D+00	-2.6080953500D-01	6.8329189010D+01
7.0000000000D-02	6.3983830450D+00	-2.6055905300D-01	8.3250448440D+01
6.0000000000D-02	6.3981325570D+00	-2.6030856570D-01	1.0354297810D+02
5.0000000000D-02	6.3978820660D+00	-2.6005807400D-01	1.3246101870D+02
4.0000000000D-02	6.3976315700D+00	-2.5980757840D-01	1.7651564680D+02
3.0000000000D-02	6.3973810710D+00	-2.5955707990D-01	2.5090306230D+02
2.0000000000D-02	6.3971305710D+00	-2.5930657900D-01	4.0121776100D+02
1.0000000000D-02	6.3968800680D+00	-2.5905607670D-01	8.5544479230D+02
9.0000000000D-03	6.3968550180D+00	-2.5903102640D-01	9.5659388780D+02
8.0000000000D-03	6.3968299680D+00	-2.5900597610D-01	1.0830742670D+03
7.0000000000D-03	6.3968049170D+00	-2.5898092580D-01	1.2457425170D+03
6.0000000000D-03	6.3967798670D+00	-2.5895587550D-01	1.4626929500D+03
5.0000000000D-03	6.3967548170D+00	-2.5893082520D-01	1.7664953360D+03
4.0000000000D-03	6.3967297660D+00	-2.5890577490D-01	2.2222892110D+03
3.0000000000D-03	6.3967047160D+00	-2.5888072460D-01	2.9820668400D+03
2.0000000000D-03	6.3966796660D+00	-2.5885567430D-01	4.5018050160D+03
1.0000000000D-03	6.3966546160D+00	-2.5883062400D-01	9.0613877230D+03
9.0000000000D-04	6.3966521100D+00	-2.5882811890D-01	1.0074650930D+04
8.0000000000D-04	6.3966496050D+00	-2.5882561390D-01	1.1341234560D+04

7.0000000000D-04	6.396647100D+00	-2.5812310890D-01	1.296970454D+04
6.0000000000D-04	6.396644595D+00	-2.5882060380D-01	1.514100403D+04
5.0000000000D-04	6.396642090D+00	-2.5881809880D-01	1.818083075D+04
4.0000000000D-04	6.396639585D+00	-2.5881559380D-01	2.274058011D+04
3.0000000000D-04	6.396637080D+00	-2.5881308870D-01	3.034017479D+04
2.0000000000D-04	6.396634575D+00	-2.5881058370D-01	4.553938276D+04
1.0000000000D-04	6.396632070D+00	-2.5880807870D-01	9.113704390D+04
9.0000000000D-05	6.396631820D+00	-2.5880782820D-01	1.012698598D+05
8.0000000000D-05	6.396631569D+00	-2.5880757770D-01	1.139358801D+05
7.0000000000D-05	6.396631319D+00	-2.5880732720D-01	1.302207638D+05
6.0000000000D-05	6.396631068D+00	-2.5880707670D-01	1.519339428D+05
5.0000000000D-05	6.396630818D+00	-2.5880682620D-01	1.823323941D+05
4.0000000000D-05	6.396630567D+00	-2.5880657560D-01	2.279300721D+05
3.0000000000D-05	6.396630317D+00	-2.5880632510D-01	3.039262031D+05
2.0000000000D-05	6.396630066D+00	-2.5880607460D-01	4.559184672D+05
1.0000000000D-05	6.396629816D+00	-2.5880582410D-01	9.118952631D+05
9.0000000000D-06	6.396629791D+00	-2.5880579910D-01	1.013223440D+06
8.0000000000D-06	6.396629766D+00	-2.5880577400D-01	1.139883662D+06
7.0000000000D-06	6.396629741D+00	-2.5880574900D-01	1.302732518D+06
6.0000000000D-06	6.396629715D+00	-2.5880572390D-01	1.519864326D+06
5.0000000000D-06	6.396629690D+00	-2.5880569890D-01	1.823848858D+06
4.0000000000D-06	6.396629665D+00	-2.5880567380D-01	2.279825655D+06
3.0000000000D-06	6.396629640D+00	-2.5880564880D-01	3.039786985D+06
2.0000000000D-06	6.396629615D+00	-2.5880562370D-01	4.559709644D+06
1.0000000000D-06	6.396629590D+00	-2.5880559870D-01	9.119477621D+06
9.0000000000D-07	6.396629588D+00	-2.5880559620D-01	1.013275939D+07
8.0000000000D-07	6.396629585D+00	-2.5880559370D-01	1.139936161D+07
7.0000000000D-07	6.396629583D+00	-2.5880559120D-01	1.302785017D+07
6.0000000000D-07	6.396629580D+00	-2.5880558870D-01	1.519916826D+07
5.0000000000D-07	6.396629578D+00	-2.5880558620D-01	1.823901358D+07
4.0000000000D-07	6.396629575D+00	-2.5880558370D-01	2.279878155D+07
3.0000000000D-07	6.396629573D+00	-2.5880558120D-01	3.039839485D+07
2.0000000000D-07	6.396629570D+00	-2.5880557860D-01	4.559762144D+07
1.0000000000D-07	6.396629568D+00	-2.5880557610D-01	9.119530122D+07
9.0000000000D-08	6.396629567D+00	-2.5880557590D-01	1.013281189D+08
8.0000000000D-08	6.396629567D+00	-2.5880557560D-01	1.139941411D+08
7.0000000000D-08	6.396629567D+00	-2.5880557540D-01	1.302790267D+08
6.0000000000D-08	6.396629567D+00	-2.5880557510D-01	1.519922076D+08
5.0000000000D-08	6.396629566D+00	-2.5880557490D-01	1.823906608D+08
4.0000000000D-08	6.396629566D+00	-2.5880557460D-01	2.279883406D+08
3.0000000000D-08	6.396629566D+00	-2.5880557440D-01	3.039844735D+08
2.0000000000D-08	6.396629566D+00	-2.5880557410D-01	4.559767394D+08
1.0000000000D-08	6.396629565D+00	-2.5880557390D-01	9.119535372D+08

MLEW ERROR -- BETA LESS THAN 0.00000001

DATA INPUT PHASE.

TOTAL NUMBER OF FAILURES = 36

TOTAL NUMBER OF SYSTEMS = 3

SYSTEM STARTING AND ENDING TIMES.

		FAILURES.				
0.	1.972D+02					
		4.300D+00	4.400D+00	1.020D+01	2.350D+01	2.380D+01
		2.640D+01	7.400D+01	7.710D+01	9.210D+01	1.972D+02
0.	1.908D+02					
		1.000D-01	5.600D+00	1.860D+01	1.950D+01	2.420D+01
		2.670D+01	4.510D+01	4.580D+01	7.570D+01	7.970D+01
		9.860D+01	1.201D+02	1.618D+02	1.806D+02	1.908D+02
0.	1.958D+02					
		8.400D+00	3.250D+01	4.470D+01	4.840D+01	5.060D+01
		7.360D+01	9.870D+01	1.122D+02	1.298D+02	1.360D+02
		1.958D+02				

ESTIMATION OF THE PARAMETERS OF THE WEIBULL INTENSITY FUNCTION.

CONSTANT NOT INVOLVING BETA: A = 3.6731896D+00

ESTIMATED BETA	FUNCTION D(B')	A - D(B')	ESTIMATED LAMBDA
1.0000000000D+01	5.1727643540+00	-1.4995747050+00	1.5274022350-22
9.0000000000D+00	5.1614702460+00	-1.4882805980+00	2.9774619750-20
8.0000000000D+00	5.1473962820+00	-1.4742066330+00	5.8030872290-18
7.0000000000D+00	5.1293520220+00	-1.4561623730+00	1.1308139340-15
6.0000000000D+00	5.1053533810+00	-1.4321637320+00	2.2031370010-13
5.0000000000D+00	5.0718289740+00	-1.3986393260+00	4.2915017950-11
4.0000000000D+00	5.0216359910+00	-1.3484463420+00	8.3578330940-09
3.0000000000D+00	4.9381078130+00	-1.2649181650+00	1.6273983530-06
2.0000000000D+00	4.7712444930+00	-1.0980548440+00	3.1681740720-04
1.0000000000D+00	4.2710460840+00	-5.9785643540-01	6.1664953750-02
9.0000000000-01	4.1599150380+00	-4.8672538910-01	1.0446158240-01
8.0000000000-01	4.0210061970+00	-3.4781654800-01	1.7695951690-01
7.0000000000-01	3.8424147990+00	-1.6922515040-01	2.9977152750-01
6.0000000000-01	3.6042995750+00	6.8890073540-02	5.0781552060-01
6.9000000000-01	3.8217088680+00	-1.4851921880-01	3.1599628650-01
6.8000000000-01	3.8003939970+00	-1.2720434790-01	3.3309918360-01
6.7000000000-01	3.7784429200+00	-1.0525327160-01	3.5112774620-01
6.6000000000-01	3.7558267200+00	-8.2637071500-02	3.7013207410-01
6.5000000000-01	3.7325146980+00	-5.9325049690-02	3.9016497840-01
6.4000000000-01	3.7084742380+00	-3.5284589480-02	4.1128212870-01
6.3000000000-01	3.6836706520+00	-1.0481003340-02	4.3354220750-01
6.2000000000-01	3.6580670150+00	1.5122634170-02	4.5700707330-01
6.2900000000-01	3.6811469200+00	-7.9572715130-03	4.3583342960-01
6.2800000000-01	3.6786151520+00	-5.4255029570-03	4.3813676040-01
6.2700000000-01	3.6760753080+00	-2.8856592200-03	4.4045226400-01
6.2600000000-01	3.6735273500+00	-3.3770160370-04	4.4278000470-01
6.2500000000-01	3.6709712400+00	2.2184088390-03	4.4512004720-01
6.2590000000-01	3.6732721070+00	-8.2458081350-05	4.4301345430-01
6.2580000000-01	3.6730167820+00	1.7286700810-04	4.4324702700-01
6.2589000000-01	3.6732465780+00	-5.6929243570-05	4.4303680610-01
6.2588000000-01	3.6732210480+00	-3.1399590080-05	4.4306015900-01
6.2587000000-01	3.6731955180+00	-5.8691208400-06	4.4308351320-01

THE FINAL ESTIMATE OF BETA = 6.2587000D-01

THE FINAL ESTIMATE OF LAMBDA = 4.4308351D-01

CONVERGENCE TO 5.8691208D-06
WHICH IS LESS THAN EPSILON = 1.0000000D-05

THE FINAL STEP SIZE IS 1.0000000D-05

CRAMER - VON MISES GOODNESS OF FIT TEST.

ESTIMATED BETA = 6.2587000D-01

NUMBER OF FAILURES = 36

UNBIASED ESTIMATE OF BETA = 5.5632889D-01

CRAMER - VON MISES STATISTIC = 1.2263085D-01

REJECT THE WEIBULL INTENSITY MODEL IF THE STATISTIC
EXCEEDS THE APPROPRIATE CRITICAL VALUE FOR M = 33

DATA INPUT PHASE.

TOTAL NUMBER OF FAILURES = 12

TOTAL NUMBER OF SYSTEMS = 1

SYSTEM STARTING AND ENDING TIMES.

FAILURES.

0. 8.000D+02

4.500D+01 4.560D+02 4.670D+02 4.770D+02 4.840D+02
4.920D+02 1.200D-01 5.640D+02 6.500D+01 7.800D+01
8.900D+01 9.910D+01

ESTIMATION OF THE PARAMETERS OF THE WEIBULL INTENSITY FUNCTION.

CONSTANT NOT INVOLVING BETA: A = 4.7045181D+00

ESTIMATED BETA	FUNCTION D(B')	A - D(B')	ESTIMATED LAMBDA
1.0000000000D+01	6.5846117280+00	-1.8800936200+00	1.1175870900-28
9.0000000000+00	6.5735006170+00	-1.8689825090+00	8.9406967160-26
8.0000000000+00	6.5596117280+00	-1.8550936200+00	7.1525573730-23
7.0000000000+00	6.5417545850+00	-1.8372364770+00	5.7220458980-20
6.0000000000+00	6.5179450610+00	-1.8134269530+00	4.5776367190-17
5.0000000000+00	6.4846117280+00	-1.7800936200+00	3.6621093750-14
4.0000000000+00	6.4346117280+00	-1.7300936200+00	2.9296875000-11
3.0000000000+00	6.3512783940+00	-1.6467602870+00	2.3437500000-08
2.0000000000+00	6.1846117280+00	-1.4800936200+00	1.8750000000-05
1.0000000000+00	5.6846117280+00	-9.8009361980-01	1.5000000000-02
9.0000000000-01	5.5735006170+00	-8.6898250870-01	2.9268485990-02
8.0000000000-01	5.4346117280+00	-7.3009361980-01	5.7109618160-02
7.0000000000-01	5.2560402990+00	-5.5152219130-01	1.1143413730-01
6.0000000000-01	5.0179450610+00	-3.1342695320-01	2.1743389910-01
5.0000000000-01	4.6846117280+00	1.9906380170-02	4.2426406870-01
5.9000000000-01	4.9896964730+00	-2.8517836560-01	2.3246531000-01
5.8000000000-01	4.9604737970+00	-2.5595568880-01	2.4853585650-01
5.7000000000-01	4.9302257630+00	-2.2570765490-01	2.6571737500-01
5.6000000000-01	4.8988974420+00	-1.9437933410-01	2.8408666820-01
5.5000000000-01	4.8664299090+00	-1.6191180170-01	3.0372584790-01
5.4000000000-01	4.8327598760+00	-1.2824176800-01	3.2472270280-01
5.3000000000-01	4.7978192750+00	-9.3301167000-02	3.4717109000-01
5.2000000000-01	4.7615348050+00	-5.7016696760-02	3.7117135540-01
5.1000000000-01	4.7238274140+00	-1.9309306110-02	3.9683078190-01
5.0000000000-01	4.6846117280+00	1.9906380170-02	4.2426406870-01
5.0900000000-01	4.7199751850+00	-1.5457077590-02	3.9949232730-01
5.0800000000-01	4.7161077910+00	-1.1589682830-02	4.0217172380-01
5.0700000000-01	4.7122251400+00	-7.7070320630-03	4.0486909100-01
5.0600000000-01	4.7083271430+00	-3.8090348540-03	4.0758454950-01
5.0500000000-01	4.7044137080+00	1.0439996760-04	4.1031822050-01
5.0590000000-01	4.7079364950+00	-3.4183875750-03	4.0785709500-01
5.0580000000-01	4.7075456940+00	-3.0275859290-03	4.0812982280-01
5.0570000000-01	4.7071547370+00	-2.6366295240-03	4.0840273290-01
5.0560000000-01	4.7067636260+00	-2.2455185690-03	4.0867582560-01
5.0550000000-01	4.7063723610+00	-1.8542528710-03	4.0894910080-01
5.0540000000-01	4.7059809400+00	-1.4628323390-03	4.0922255880-01
5.0530000000-01	4.7055893650+00	-1.0712568820-03	4.0949619960-01
5.0520000000-01	4.7051976340+00	-6.7952640610-04	4.0977002350-01
5.0510000000-01	4.7048057490+00	-2.8764082040-04	4.1004403040-01

5.0500000000D-01	4.7044137088D+00	1.0439996760-04	4.1031822050-01
5.0509000000D-01	4.7047665520+00	-2.4844372720-04	4.1007144120-01
5.0508000000D-01	4.7047273530+00	-2.0924508190-04	4.1009885380-01
5.0507000000D-01	4.7046881530+00	-1.7004488440-04	4.1012626820-01
5.0506000000D-01	4.7046489510+00	-1.3084313460-04	4.1015368450-01
5.0505000000D-01	4.7046097480+00	-9.1639832430-05	4.1018110260-01
5.0504000000D-01	4.7045705430+00	-5.2434977770-05	4.1020852250-01
5.0503000000D-01	4.7045313360+00	-1.3228570530-05	4.1023594420-01
5.0502000000D-01	4.7044921280+00	2.5979389380-05	4.1026336780-01
5.0502900000D-01	4.7045274160+00	-9.3078444090-06	4.1023868650-01

THE FINAL ESTIMATE OF BETA = 5.0502900D-01

THE FINAL ESTIMATE OF LAMBDA = 4.1023869D-01

CONVERGENCE TO 9.3078444D-06
WHICH IS LESS THAN EPSILON = 1.0000000D-05

THE FINAL STEP SIZE IS 1.0000000D-06

CRAMER - VON MISES GOODNESS OF FIT TEST.

ESTIMATED BETA = 5.0502900D-01

NUMBER OF FAILURES = 12

UNBIASED ESTIMATE OF BETA = 4.62943250-01

CRAMER - VON MISES STATISTIC = 1.4645211D-01

REJECT THE WEIBULL INTENSITY MODEL IF THE STATISTIC
EXCEEDS THE APPROPRIATE CRITICAL VALUE FOR M = 12

DATA INPUT PHASE.

PROGRAM RUN ENDS NORMALLY.

Next page is blank.

DISTRIBUTION LIST

No. of Copies	Organization
12	Commander Defense Documentation Center ATTN: TCA Cameron Station Alexandria, VA 22314
1	Commander US Army Materiel Development and Readiness Command ATTN: DRCCP 5001 Eisenhower Avenue Alexandria, VA 22333
1	Commander US Army Materiel Development and Readiness Command ATTN: DRCDE-F 5001 Eisenhower Avenue Alexandria, VA 22333
1	Commander US Army Materiel Development and Readiness Command ATTN: DRCRE-I 5001 Eisenhower Avenue Alexandria, VA 22333
1	Commander US Army Materiel Development and Readiness Command ATTN: DRCBSI-L 5001 Eisenhower Avenue Alexandria, VA 22333
1	Commander US Army Materiel Development and Readiness Command ATTN: DRCPA-S 5001 Eisenhower Avenue Alexandria, VA 22333

DISTRIBUTION LIST

No. of Copies	Organization
1	Commander US Army Materiel Development and Readiness Command ATTN: DRCQA 5001 Eisenhower Avenue Alexandria, VA 22333
1	Commander US Army Materiel Development and Readiness Command ATTN: DRCBSI-D 5001 Eisenhower Avenue Alexandria, VA 22333
1	Commander US Army Materiel Development and Readiness Command ATTN: DRCDE-R 5001 Eisenhower Avenue Alexandria, VA 22333
1	Commander US Army Materiel Development and Readiness Command ATTN: DRCDE-D 5001 Eisenhower Avenue Alexandria, VA 22333
1	Commander US Army Armament Materiel Readiness Command ATTN: DRSAR-SA Rock Island, IL 61299
1	Commander Rock Island Arsenal ATTN: Tech Lib Rock Island, IL 61299
1	Commander US Army Armament Research and Development Command Dover, NJ 07801

DISTRIBUTION LIST

No. of Copies	Organization
1	Commander Harry Diamond Laboratories ATTN: DELHD-SAB 2800 Powder Mill Road Adelphi, MD 20783
1	Commander US Army Test & Evaluation Command ATTN: STEDP-MT-L Dugway Proving Ground, MD 84022
1	Commander US Army Aviation Research and Development Command ATTN: DRDAV-BC P. O. Box 209 St. Louis, MO 63166
1	Commander US Army Electronics Research and Development Command ATTN: DRDEL-SA Fort Monmouth, NJ 07703
2	Director US Army TRADOC Systems Analysis Activity ATTN: ATAA-SL ATAA-T White Sands Missile Range, NM 88002
1	Commander US Army Missile Command ATTN: DRSMI-C Redstone Arsenal, AL 35809
1	Commander US Army Troop Support and Aviation Materiel Readiness Command ATTN: DRSTS-F 4300 Goodfellow Blvd St. Louis, MO 63120

DISTRIBUTION LIST

No. of Copies	Organization
1	Commander US Army Tank-Automotive Research and Development Command ATTN: DRDTA-UL (Tech Lib) DRDTA-V Warren, MI 48090
1	Commander US Army Mobility Equipment Research and Development Command ATTN: DRDME-O Fort Belvoir, VA 22060
1	Commander US Army Natick Research and Development Command ATTN: DRDNA-O Natick, MA 01760
2	Chief Defense Logistics Studies Information Exchange US Army Logistics Management Center ATTN: DRXMC-D Fort Lee, VA 23801
1	Commander US Army Concepts Analysis Agency 8120 Woodmont Avenue Bethesda, MD 20014
1	Commander US Army Logistics Center ATTN: ATCL-MER (Mr. H. Burnette) Fort Lee, VA 23801

Aberdeen Proving Ground

Cdr, USATECOM
ATTN: DRSTE, DRSTE-CS-A
Bldg 314

Dir, BRL, ATTN: STINFO BR, Bldg 305

Director, BRL, Bldg 328

Director, HEL, Bldg 520